CHROOT-03

Close file descriptors when using chroot()

Sean Barnum, Cigital, Inc. [vita[1]]

2007-03-19

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 5405 bytes

Attack Category	• Privilege Exploitation
Vulnerability Category	• Privilege escalation problem • Indeterminate File/Path
Software Context	• Security • Authorization
Location	
Description	Close file descriptors when using chroot(). The chroot() function establishes a virtual root directory for the owning process. This may be used to limit the amount of file system access a potential hacker could use if he or she gained control of the process. Programs like ftp and httpd commonly make use of this function. The chroot() function isolates a small part of the file system. However, any open file descriptors may still point outside the isolated area, giving an attacker access to areas outside the desired sandbox. Use of chroot is desirable, but should also be a flag to indicate that one needs to ensure that related security issues are addressed.
APIs	**FunctionName** / **Comments** chroot fclose — use fclose to clean up file descriptors after chroot() calls
Method of Attack	An attacker who gains control of the process by exploiting another vulnerability can potentially access files that chroot was supposed to protect if open file descriptors for those files are available.
Exception Criteria	
Solutions	**Solution Applicability** / **Solution Description** / **Solution Efficacy**

1. http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)

	Whenever chroot is used.	Loop through and close all open file descriptors after (or before) the chroot() call.	Effective at eliminating access to open files, but must ensure that other chroot issues are addressed also.
Signature Details	int chroot(const char *)		

Examples of Incorrect Code	`[...]`
	`char path[] = "/usr/sandbox";`
	`chroot(path);`
	`[...]`
	`/* Continuing without changing user ID is a security risk because running as root. */`		

Examples of Corrected Code	`[...]`
	`char path[] = "/usr/sandbox";`
	`fclose(anOpenFileDescriptor); /* Should not leave file descriptors open. */`
	`if (chroot(path)) exit(1); /* Should check return value. */`
	`chdir("/"); /* Must do this or chroot() won't have intended effect */`
	`setegid(ogid); /* Should change group ID */`
	`seteuid(ouid); /* Should change user ID */`
	`[...]`
	`/* Now can safely continue */`		

Source Reference	• Viega, John & McGraw, Gary. Building Secure Software: How to Avoid Security Problems the Right Way. Boston, MA: Addison-Wesley Professional, 2001, ISBN: 020172152X, pg. 205.		
Recommended Resource	• chroot man page[2]		
Discriminant Set	**Operating System**	• UNIX (All)	
	Languages	• C • C++	

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. mailto:copyright@cigital.com